

I CLAIM:

- 1) A natural language processing apparatus for translating natural language into a formal language executable on a programmable device, said system comprising,
 - a) memory for storing data;
 - b) a data processor;
 - c) an input device for presenting natural language text to said system;
 - d) a text parser for partitioning said text into a sequence of sequences of string of characters or pretokens;
 - e) a lexicon for storing lexical terms as token associated with lexical type and reference data;
 - f) a lexical type assignment process for assigning lexical types to pretokens by comparison to terms in the lexicon;
 - g) a lexical insertion processor for inserting terms into the lexicon under specific control;
 - h) a control processor for invoking lexical insertions under the condition that a pretoken is not recognized as a lexical token;
 - i) a type contextualization processor by which refined lexical types may be reassigned to tokens depending on syntactic context;
 - j) a type reduction matrix;
 - k) a term reduction processor which uses said type reduction matrix to determine proper syntactic dependencies between tokens in a sentence;
 - l) a term inversion processor for constructing chains of syntactic dependencies among lexical terms in an expression and for determining the proper dependencies between those chains;
 - m) a syntactic tree generation processor for constructing syntactic trees representing the syntactic structure of each processed expression;
 - n) a syntactic algebra comprising syntactic terms formally representing processed expressions;

- o) a syntactic representation processor for constructing syntactic terms to represent the formal syntactic structure of processed expressions;
- p) a semantic algebra comprising semantic objects as formal references of appropriate terms in the syntactic algebra;
- q) a semantic representation processor for associating internal semantic object references with terms in the syntactic algebra;
- r) a semantic tensor algebra comprising correlated pairs of syntactic algebraic terms and their semantic object representations;
- s) a formal representation processor for associating appropriate internal formal models with terms in the semantic tensor algebra;
- t) a formal interpretation processor for transforming terms in the syntactic algebra into equivalent expressions in an internal formal language;
- u) an external representation processor for associating external operational environments with internal formal models;
- v) an external interpretation processor for translating expressions in an internal formal language into equivalent formal expressions executable into appropriate external operational environments.

2) A method for translating natural language into a formal language executable on a programmable device, said method comprising the steps of:

- a) receiving natural language text;
- b) parsing said text into a sequence of sequences of pretokens;
- c) recognizing pretokens as tokens in the lexicon;
- d) inserting new terms into the lexicon under specific control;

- e) assigning types to pretokens to form lexical terms for further syntactic processing;
- f) reassigning lexical types to tokens based on syntactic context;
- g) correlating terms occurring in a set of expressions in order to replace indirect references by appropriate direct references;
- h) establishing syntactic dependencies between terms in an expression through a process of term reduction;
- i) constructing chains of syntactic dependencies and determining dependencies between those chains, by a process of term inversion;
- j) generating syntactic trees which represent the syntactic structures of said processed expressions;
- k) representing said processed expressions as terms in a syntactic algebra;
- l) representing terms in the syntactic algebra as objects in the semantic algebra;
- m) combining objects in the semantic algebra by means of a semantic product on pairs of semantic objects to form more complex semantic objects;
- n) representing correlated syntactic algebraic terms and semantic objects as terms in a semantic tensor algebra;
- o) representing terms in the semantic tensor algebra as internal formal models;
- p) transforming terms in the syntactic algebra into equivalent expressions in an internal formal language;
- q) associating external operation environments with internal formal models; and
- p) translating expressions of the internal formal language into equivalent formal expressions executable in an external operational environment.

3) In a natural language processing apparatus for translating natural language into a formal language executable on a programmable device, wherein said system includes processing means; input means for presenting natural language text to said system; a lexicon of terms; a text parser which partitions expressions into sequences of sequences of pretokens; a type assignment process for assigning syntactic types to pretokens by comparison to lexical terms in the lexicon and determining their status as tokens; a type contextualization process for reassigning lexical types to tokens based on syntactic context, a term correlation process for correlating terms occurring in a set of expressions in order to replace indirect references by direct references, said system comprising

- a) a type reduction matrix;
- b) a term reduction processor that uses the type reduction matrix to determine proper syntactic dependencies between tokens in an expression;
- c) a term inversion processor for constructing chains of syntactic dependencies among lexical terms in an expression and for determining the proper dependencies between those chains;
- d) a syntactic tree generation processor for constructing syntactic trees representing the syntactic structures of expressions;
- e) a syntactic algebra comprising syntactic terms formally representing processed expressions;
- f) a syntactic representation processor for constructing syntactic algebraic terms representing processed expressions;
- g) a semantic object algebra comprising semantic objects as internal references of terms in the syntactic algebra;
- h) a semantic product processor by which objects in the in the semantic object algebra are combined to form more complex semantic objects;

- i) a semantic representation processor by which internal semantic algebraic objects representing terms in the syntactic algebra are constructed;
- j) a semantic tensor algebra comprising correlated syntactic terms and semantic objects;
- k) a formal representation processor by which internal formal models are associated with terms in the semantic tensor algebra;
- l) a formal interpretation processor by which syntactic algebraic terms are transformed into equivalent expressions in an internal formal language;
- m) a semantic product processor by which objects in the semantic algebra are combined to form more complex semantic objects;
- n) an external representation processor by which external operational environments are associated with internal formal models; and
- o) an external interpretation processor by which expressions in an internal formal language are translated into equivalent formal expressions executable in an external environment;

4) A software system for translating natural language into a formal language executable on a programmable device, wherein said system includes processing means; input means for presenting natural language text to said system; a lexicon of terms; a text parser which partitions natural language texts into sequences of sequences of pretokens; a type assignment process for assigning syntactic types to pretokens by comparison to lexical terms in the lexicon and determining their status as tokens; a type contextualization process for reassigning lexical types to tokens based on syntactic context; a term correlation process for correlating terms occurring in a set of expressions in order to replace indirect references by direct references,

- a) a type reduction matrix;
- b) a term reduction process which uses the type reduction matrix to determine proper syntactic

dependencies between tokens in an expression;

c) a term inversion process for constructing chains of syntactic dependencies among lexical terms in an expression and for determining the proper dependencies between those chains;

d) a syntactic tree generation process by which syntactic trees representing the syntactic structures of expressions are constructed;

e) a syntactic algebra comprising syntactic terms formally representing processed expressions;

f) a syntactic representation process by which syntactic algebraic terms representing processed expressions are constructed;

g) a semantic object algebra comprising semantic objects as internal references of term in the syntactic algebra;

h) a semantic object algebra comprising semantic objects as formal references of terms in the syntactic algebra;

i) a semantic representation process by which internal semantic algebraic objects representing appropriate terms in the syntactic algebra are constructed;

j) a semantic product process by which objects in the semantic algebra are combined to form more complex semantic objects;

k) a formal representation process by which internal formal models object references are associated with terms in the semantic tensor algebra;

j) a formal interpretation process by which syntactic algebraic terms are transformed into equivalent expressions in an internal formal language;

l) an external representation process by which appropriate external operation environments are associated with internal formal models; and

l) an external interpretation process by which expressions in an internal formal language are translated into equivalent formal expressions executable in an external operational environment.

5. A software system for a data processing device used in translating natural language into executable expressions in a formal language, wherein said data processing device includes a data processor and memory; input means for presenting natural language text to said system; a lexicon of terms; a text parser which partitions natural language texts into sequences of sequences of pretokens; a type assignment processor for assigning syntactic types to pretokens by comparison to lexical terms in the lexicon and determining their status as tokens; a type contextualization processor for reassigning lexical types to tokens based on syntactic context; a term correlation processor for correlating terms occurring in a set of expressions in order to replace indirect references by direct references; said software system comprising,

- a) a type reduction matrix for processing said expressions;
- b) a term reduction processor that uses the type reduction matrix to determine proper syntactic dependencies between tokens in an expression;
- c) a term inversion processor for constructing chains of syntactic dependencies among lexical terms in an expression and for determining the proper dependencies between those chains;
- d) a syntactic tree generation processor by which syntactic trees representing the syntactic structures of expressions are constructed;
- e) a syntactic algebra comprising syntactic terms formally representing said processed expressions;
- f) a syntactic representation processor by means of which syntactic algebraic terms representing processed expressions are constructed;
- g) a semantic object algebra comprising semantic objects as internal references of terms in the syntactic algebra;
- h) a semantic representation processor by which internal semantic algebraic objects representing

terms in the syntactic algebra are constructed;

i) a semantic product processor by which objects in the semantic algebra are combined to form more complex semantic objects;

j) a formal representation processor by which internal formal models are associated with terms in the semantic tensor algebra;

k) a formal interpretation processor by which syntactic algebraic terms are transformed into equivalent expressions in an internal formal language;

l) an external representation processor by which external operational environments are associated with internal formal models; and

m) an external interpretation processor by which expressions in an internal formal language are translated into equivalent formal expressions executable in an external operational environment.

6. A system as in claim 5 further including a protocol for connecting the output of said translating processor to digitally responsive machines and other data responsive devices.

7. A system as in claim 5 further including a protocol by means of which

a) selected ones of said internal formal models are associated with terms in said semantic tensor algebra;

b) syntactic algebraic terms are transformed into equivalent expressions in the internal formal language;

c) selected external operational environments are associated with selected formal models; and

d) expressions in the internal formal language are translated into equivalent formal expressions executable in an external operational environment.

8. A system as in claim 5 further comprising,

a) a lexical insertion processor for inserting lexical terms into the lexicon under user control

whereby said lexicon can be expanded and refined; and

b) a controller for invoking lexical insertions under the condition that a pretoken is not recognized as a lexical token.

9. A system as in claim 7 further including a process control for inserting external lexical information to said lexicon to enable the system to learn new lexical information including vocabulary and associated lexical type and reference relations.

10. A data processing system for translating a natural language into a language executable as a formal machine language comprising, in combination,

a) input devices for inputting a natural language text to said system;

b) text processing components for providing an output comprising a sequence of preexpressions based on said text;

c) a syntactic processing component receiving said preexpressions and providing a sequence of syntactic complexes;

d) semantic processing components for receiving said sequence of syntactic complexes and providing a sequences of formal expressions; and

e) external processing components for providing a sequence of executable expressions to an external operational environment based on said formal expressions.

11. A method of translating a natural language into a language executable as a formal or machine language comprising the steps of ,

- and

- | Parameter | Unit | Value | Standard Error | t-Statistic | p-Value |
|-------------------------------|-----------------------------------|--------|----------------|-------------|---------|
| Intercept | | 1.0000 | 0.0000 | 1.0000 | 0.0000 |
| Age | Years | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age squared | Years squared | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age cubed | Years cubed | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quartic | Years quartic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quintic | Years quintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age sextic | Years sextic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age septic | Years septic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age octic | Years octic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age nonic | Years nonic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age decic | Years decic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age undecic | Years undecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age duodecic | Years duodecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age tredecic | Years tredecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quattuordecic | Years quattuordecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quindecic | Years quindecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age sexdecic | Years sexdecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age septendecic | Years septendecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age octodecic | Years octodecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age novemdecic | Years novemdecic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age vigintic | Years vigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age unvigintic | Years unvigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age bivigintic | Years bivigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age trivigintic | Years trivigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quadravigintic | Years quadravigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quinquavigintic | Years quinquavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age sexavigintic | Years sexavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age septuavigintic | Years septuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age octuavigintic | Years octuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age nonuavigintic | Years nonuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age decuavigintic | Years decuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age undecuavigintic | Years undecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age duodecuavigintic | Years duodecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age tredecuavigintic | Years tredecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quattuordecuavigintic | Years quattuordecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quindecuavigintic | Years quindecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age sexdecuavigintic | Years sexdecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age septendecuavigintic | Years septendecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age octodecuavigintic | Years octodecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age novemdecuavigintic | Years novemdecuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age viginticuavigintic | Years viginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age unviginticuavigintic | Years unviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age biviginticuavigintic | Years biviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age triviginticuavigintic | Years triviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quadraviginticuavigintic | Years quadraviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age quinquaviginticuavigintic | Years quinquaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age sexaviginticuavigintic | Years sexaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age septuaviginticuavigintic | Years septuaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age octuaviginticuavigintic | Years octuaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age nonuaviginticuavigintic | Years nonuaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age decuaviginticuavigintic | Years decuaviginticuavigintic | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Age undecuaviginticuavigintic | Years undecuaviginticuavigintic</ | | | | |